Kapitola: Primitívna rekurzia

1.1 Motivácia

Matematických funkcií je nespočítateľne veľa. Dokonca už unárnych funkcií na $\mathbb{N}$ je nespočítateľne veľa. (Ľahký dôkaz úpravou Cantorovej diagonalizácie.)

Tým pádom ale väčšina z nich pre nás nie je zaujímavá – nikdy nikde takéto funkcie nestretneme. Istá zaujímavosť začína až u funkcií, ktoré vieme exaktne definovať.

Samotná definícia však ešte nemusí byť všetko. Uvažujme napríklad funkciu:

$$fives(x) = \begin{cases} 1 & \leftarrow \text{v desatinnom rozvoji } \pi \text{ sa niekde nachádza } x \text{ pätiek za sebou} \\ 0 & \leftarrow \text{inak} \end{cases}$$

Toto je síce exaktná definícia, jasne popisujúca práve jednu funkciu – je nám ale nanič. Nehovorí nám v podstate nič o tom, ako túto funkciu vypočítať. Je dokonca otvorený problém či je $fives$ identicky rovná jednej.

Všimnite si tiež rozdiel medzi „(na základe definície) nevieme funkciu vyhodnotiť" a „neexistuje algoritmus ktorý by túto funkciu vyhodnocoval". To prvé je v tomto prípade (zatiaľ) pravda, to druhé však nie. Totiž určite existuje triviálny program, ktorý funkciu $fives$ ráta. My len nevieme, či je to program:

```
return 1;
```

alebo jeden z programov tvaru:

```
if (vstup < K) then return 1 else return 0;
```

Pekné funkcie by teda boli také, u ktorých je definícia priamo „návodom" na ich počítanie.

Ako takéto definície robiť systematicky? Pre úvod: Ako ich robiť systematicky pre funkcie na $\mathbb{N}$?

Pekná funkcia je napríklad $f(x, y) = x^y$. Ako to vyzerá, keď ju vyhodnocujeme? Hodnotu x^y dostaneme tak, že zoberieme 1ku a y-krát ju vynásobíme hodnotou x.

Čo znamená „vynásobíme hodnotu a hodnotou b"? Zoberieme nulu a b-krát k nej pripočítame a.

A čo znamená „pripočítame"?

Takto sa môžeme na zložité procesy vyhodnocovania funkcií dívať ako na veľa malých, jednoduchých krokov. Otázka znie, kde už sme nútení prestať, čo sa už nedá ďalej zjednodušiť?

Taktiež nás tento príklad môže inšpirovať k voľbe nástroja na veľmi jednoduchý popis mnohých funkcií: rekurziu. Napr. „x umocnené na y" môžeme definovať pomocou predpisu „x umocnené na $n+1$" je „x umocnené na n" krát x. (Plus samozrejme potrebujeme začiatočné podmienky, v tomto prípade: „x umocnené na 0" je 1.)

1.2 Definícia

Trieda $PREC$ primitívne rekurzívnych funkcií je definovaná nasledovne:

1. Nulárna funkcia z taká, že $z() = 0$, je v $PREC$.

   (Formálnejšie: z je funkcia s 0 vstupmi a 1 výstupom, ktorý je rovný nule.)

2. Unárna funkcia s definovaná predpisom $\forall x : s(x) = x + 1$ je v $PREC$.

   Túto funkciu voláme nasledovník (successor).

3. Pre každé $1 \le k \le n$ je v $PREC$ n-árna funkcia P^n_k definovaná $\forall x_1, \dots, x_n : P^n_k(x_1, \dots, x_n) = x_k$.

   Tieto funkcie voláme projekcie. Špeciálne funkciu P^n_k voláme „k-ta projekcia z n".

4. Ak $g \in PREC$ je k-árna funkcia a $f_1, \dots, f_k \in PREC$ sú m-árne funkcie, potom m-árna funkcia $h(x_1, \dots, x_m) = g(f_1(x_1, \dots, x_m), \dots, f_k(x_1, \dots, x_m))$ je v $PREC$.

   Hovoríme, že h vzniká kompozíciou pôvodných funkcií. Zapisujeme to $h \equiv COMP[g, f_1, \dots, f_k]$.

5. Ak $f \in PREC$ je k-árna a $g \in PREC$ je $(k+2)$-árna funkcia. Potom $h \in PREC$, kde $h(0, x_1, \dots, x_k) = f(x_1, \dots, x_k)$ a $h(s(n), x_1, \dots, x_k) = g(h(n, x_1, \dots, x_k), n, x_1, \dots, x_k)$.

   Hovoríme, že h vzniká z f a g primitívnou rekurziou. Zapisujeme to $h \equiv PR[f, g]$.

6. V $PREC$ nie sú žiadne iné funkcie.

Ekvivalentne, $PREC$ je najmenšia trieda funkcií, ktorá obsahuje funkcie z bodov 1-3 a je uzavretá vzhľadom na kompozíciu a na primitívnu rekurziu.

Všimnite si, že operácia primitívnej rekurzie je „pekná". Vyhodnotenie konkrétnej hodnoty funkcie h môže vyžadovať vyhodnotenie h aj pre iné hodnoty. Môžeme si však všimnúť, že hodnota prvého parametra odovzdávaného funkcii h pri každom kroku klesá, preto náš výpočet určite skončí po vyhodnotení h pre konečne veľa vstupov.

Všetky funkcie v $PREC$ sú totálne: dôkaz indukciou podľa definície.

Pre formalistov

Pre zarytých formalistov tu máme poznámku k nulárnym funkciám: Formálne si môžeme $\mathbb{N}^k$ definovať ako množinu všetkých k-tic prirodzených čísel. Špeciálne teda $\mathbb{N}^0$ **nie je** prázdna množina – je to množina obsahujúca všetky 0-tice prirodzených čísel. Teda množina obsahujúca jeden prvok: prázdnu 0-ticu prirodzených čísel. Ak si tú označíme napr. symbolom ε, tak je $\mathbb{N}^0 = \{\varepsilon\}$.

Nulárne funkcie, ako napr. funkcia z (zero), sú potom priamočiaro funkcie z $\mathbb{N}^0$ do $\mathbb{N}$; konkrétne z vstupu ε priradí výstup 0.

Toto nás však nijak extra netrápi, na tento level formalizmu sa nepotrebujeme znížiť. Spokojne môžeme s nulárnymi funkciami pracovať ako s funkciami, ktoré nemajú žiadne vstupy a vracajú konštantu.

1.3 Príklady

1.3.1 Konštanty

Konštanty, presnejšie nulárne funkcie, vieme vyrobiť postupnou kompozíciou veľa successorov a nuly.

Formálne môžeme napríklad nulárnu funkciu j takú, že $j() = 1$, vyrobiť nasledovne: $j \equiv Comp[s, z]$. (Slovne: j vzniká kompozíciou, pri ktorej do funkcie s dosadíme funkciu z.)

Unárnu nulu vieme rekurzívne definovať napr. nasledovne: $z_1(0) = z = 0$ a $\forall n : z_1(n+1) = z_1(n)$.
Podľa tejto definície ju vieme formálne zostrojiť: $z_1 \equiv PR[z, P_1^2]$.
Ostatné unárne konštanty dostaneme postupnou kompozíciou successorov a unárnej nuly.

Konštantné nuly s vyššou aritou sa dá vyrobiť viacerými spôsobmi, napr.:

- Rovnako, ako sme vyrobili z_1 zo z, vieme následne vyrobiť primitívnou rekurziou binárnu nulu z_2 zo z_1, a tak ďalej.
- Vieme priamo vyrobiť k-árnu nulu (pre $k > 1$) primitívnou rekurziou zo z_1 a P_1^{k+1}.
- Ide to aj bez primitívnej rekurzie: na výrobu z_k stačí kompozíciou do z_1 dosadiť P_1^k.

Iné konštanty s vyššou aritou opäť dostaneme postupnou kompozíciou successorov a nuly správnej arity.

1.3.2 Identita

Identitu máme samozrejme v našej množine projekcií, je to $i \equiv P_1^1$.

1.3.3 Sčítanie

Intuitívne: sčítanie môžeme definovať nasledovne:

$$add(0, y) = y$$
$$add(x + 1, y) = add(x, y) + 1$$

Podľa tejto intuitívnej definície vyrobíme formálnu, pomocou operácie primitívnej rekurzie. Na jej použitie potrebujem mať dve funkcie. Funkcia f popisujúca základný prípad má v našom prípade pre ľubovoľný vstup y vrátiť výstup y, je to teda identita.

Funkcia g, ktorá počíta rekurzívny krok, dostane vstupy $add(x, y)$, x a y a má vrátiť výstup $add(x, y) + 1$. Teda ide o funkciu $g(u, v, w) = u + 1$. Takáto funkcia je zjavne primitívne rekurzívna, lebo vzniká kompozíciou s a P^3_1.

Formálne teda $add \equiv PR[P^1_1, Comp[s, P^3_1\ ]\ ]$.

1.3.4 Násobenie

Intuitívne: násobenie môžeme definovať nasledovne:

$$mul(0, y) = 0$$
$$mul(x + 1, y) = mul(x, y) + y$$

Formálne teda $mul \equiv PR[z_1, Comp[add, P^3_1, P^3_3]]$.

Fungovalo by aj $mul \equiv PR[z_1, Comp[add, P^3_3, P^3_1]]$. (Ide o iný popis, t.j., výpočet funkčnej hodnoty by prebiehal ináč. Ale oba popisy popisujú ten istý matematický objekt – funkciu počítajúcu „x krát y“.)

Pomocou sčítania sme práve definovali násobenie. Pomocou násobenia teraz môžeme rovnako definovať umocňovanie, pomocou neho poschodové mocniny, a tak ďalej. To, čo definujeme z poschodových mocnín, už rastie fakt dosť rýchlo.

1.3.5 Predecessor

Pri definícii predecessora máme problém so vstupom 0, keďže sa hráme len s prirodzenými číslami, a teda nemáme hodnotu -1. Dohodneme sa preto, že predecessorom nuly bude opäť nula. Použitie predecessora môžeme teda čítať „zmenši, ak sa to dá“.

Intuitívna rekurzívna definícia:

$$p(0) = 0$$
$$p(x + 1) = x$$

Podľa nej ľahko napíšeme definíciu formálnu: $p \equiv PR[z, P^2_2]$.

1.3.6 Odčítanie

Keďže primitívnu rekurziu vieme robiť len podľa prvého parametra, odčítanie spravíme na dva kroky. Najskôr spravíme funkciu, ktorá bude mať parametre v opačnom poradí: $m(y, x)$, ktorá „od x odčíta y“. Prečo tie úvodzovky? Preto, že opäť máme problém so zápornými číslami, teda so situáciami, kedy $y > x$. Pre tieto sa dohodneme, podobne ako u predecessora, že $m(y, x)$ bude vracať 0. Matematicky môžeme teda funkciu m definovať nasledovne: $\forall x, y : m(y, x) = \max(0, x - y)$.

(Vyššie uvedená definícia je síce matematicky korektná, treba si ale uvedomiť, že nejde o definíciu v rámci formalizmu primitívnej rekurzie. Vyššie uvedený zápis nám teda nestačí na to, aby sme o funkcii m usúdili, že je primitívne rekurzívna.)

Funkcia m spĺňa nasledovnú rekurzívnu definíciu (pričom p je predecessor):

$$m(0, x) = x$$
$$m(y + 1, x) = p(m(y, x))$$

Podľa nej teda zostrojíme m v našom formalizme: $m \equiv PR[P^1_1, Comp[p, P^3_1]]$.

No a teraz z m zostrojíme funkciu sub, ktorá už má parametre v očakávanom poradí. Na prehodenie parametrov použijeme kompozíciu s vhodnými projekciami: $sub \equiv Comp[m, P^2_2, P^2_1]$.

1.4 Meta-veta o kompozícii

Ako už naznačuje konštrukcia funkcie *sub*, projekcie sa dajú používať na prehadzovanie poradia parametrov a menenie ich počtu.

Keď sme definovali kompozíciu, spravili sme definíciu veľmi striktne: všetky funkcie f_i museli mať presne rovnakú aritu a dostávali presne tie isté vstupy. V súčasnej dobe sme však zvyknutí pojmom „kompozícia" označovať všeobecnejšie skladanie funkcií. Napríklad aj o takejto funkcii φ hovoríme, že vzniká kompozíciou (zložením) funkcií použitých na pravej strane:

$$\forall x, y, z : \varphi(x, y, z) = \psi(\alpha(x), \beta(y, y, z), \gamma(z, x))$$

Meta-veta o kompozícii: Každá funkcia, ktorú vieme zapísať ako takúto všeobecnú kompozíciu primitívne rekurzívnych funkcií, je tiež primitívne rekurzívna.

Dôkaz (príkladom): Na vyššie uvedenej funkcii φ predvedieme, ako všeobecnú kompozíciu „preložiť" do povolenej podoby. Jediné, čo spravíme, je úprava funkcií α, β a γ do podoby, v ktorej bude každá z nich brať na vstupe všetky tri parametre funkcie φ v správnom poradí.

V našom prípade teda vyrobíme nové funkcie: $\alpha' \equiv Comp[\alpha, P^3_1]$, $\beta' \equiv Comp[\beta, P^3_2, P^3_2, P^3_3]$ a $\gamma' \equiv Comp[\gamma, P^3_3, P^3_1]$.

Keďže funkcie α', β' a γ' vznikli z primitívne rekurzívnych funkcií kompozíciou, sú tiež primitívne rekurzívne. A následne je teda primitívne rekurzívna aj funkcia $\varphi \equiv Comp[\psi, \alpha', \beta', \gamma']$.